

Tvheadend - Feature #4366

Create bouquet for all network types / per network (was: map all services through json api...)

2017-05-11 12:03 - edit4ever !

Status:	Fixed	Start date:	2017-05-11
Priority:	Normal	Due date:	
Assignee:		% Done:	100%
Category:	API	Estimated time:	0.00 hour
Target version:	4.2		
Description			
The ability to start the map all services function was removed from the json api.			
In the USA - we need a simple way to scan muxes (which does exist in the json api) and then have the new services automatically mapped to channels. With all of our ATSC 3.0 changes coming, stations are going to be changing frequencies. I would like to trigger this from my frontend without having to go through the web interface.			
Could this "start" function for the mapping services api be put back in?? If not - what is the method to map services through the current json api options?			

Associated revisions

Revision e76171c5 - 2017-05-12 14:32 - Jaroslav Kysela

mpepts network: create network bouquet for all mpepts networks, fixes #4366

History

#1 - 2017-05-11 16:53 - Jaroslav Kysela

- Target version set to 4.2

Check the webui - you may send the service uuid list to the /api/service/mapper/load .

Anyway, it may be better to manage a bouquet of all services in network. Then users may enable the automatic service mapping through the bouquets. I'll work on it.

#2 - 2017-05-11 17:37 - edit4ever !

Thanks for looking into this! I'm striggling with sending the uuid to the /api/service/mapper/load - perhaps I'm not formatting the request properly?

I've tried the following:

<http://192.168.80.67:9981/api/service/mapper/load?uuid=%220bb59670b3aa5d2e412d15ea82fc3c71%22.%220e63e4f90edc546d4d4b7c282f1326b9%22.%220d1c8fa4205a5b8c6f30dde04a098652%22>

<http://192.168.80.67:9981/api/service/mapper/load?uuid={%220bb59670b3aa5d2e412d15ea82fc3c71%22.%220e63e4f90edc546d4d4b7c282f1326b9%22.%220d1c8fa4205a5b8c6f30dde04a098652%22}>

[http://192.168.80.67:9981/api/service/mapper/load?uuid=\[%220bb59670b3aa5d2e412d15ea82fc3c71%22.%220e63e4f90edc546d4d4b7c282f1326b9%22.%220d1c8fa4205a5b8c6f30dde04a098652%22\]](http://192.168.80.67:9981/api/service/mapper/load?uuid=[%220bb59670b3aa5d2e412d15ea82fc3c71%22.%220e63e4f90edc546d4d4b7c282f1326b9%22.%220d1c8fa4205a5b8c6f30dde04a098652%22])

But no change. I'm sure it's something simple - so thank in advance for providing the correct format for the list.

#3 - 2017-05-11 21:02 - Jaroslav Kysela

Sorry, it should be:

```
http://localhost:9981/api/service/mapper/save
```

JSON parameters:

```
node:{"services":["cb86b1f9fd5244f522b4792050f34773","501419945e4058bf11b5756491ed5f16"],"encrypted":true,"merge_same_name":false,"check_availability":false,"type_tags":true,"provider_tags":false,"network_tags":false}
```

You can use any browser's debugger (network view) to check which values the webui sends to tvh.

#4 - 2017-05-12 16:32 - Jaroslav Kysela

- Status changed from New to Fixed

- % Done changed from 0 to 100

Applied in changeset [tvheadend/e76171c573492fe52d56bba25c768db901e017f7](https://github.com/xbmc/xbmc/commit/tvheadend/e76171c573492fe52d56bba25c768db901e017f7).

#5 - 2017-05-12 16:34 - Jaroslav Kysela

You can (since v4.3-60-ge76171c) create a bouquet per network now. It means that if you check 'Create bouquet' flag in the network settings, you can configure the auto channel mapping inside the created bouquet for your whole network. No external tool is required now.

#6 - 2017-05-12 16:37 - Jaroslav Kysela

- Subject changed from *Map all services through json api...* to *Create bouquet for all network types / per network (was: map all services through json api...)*

#7 - 2017-05-13 15:42 - edit4ever !

Awesome - thanks for this. I will test the new create bouquet feature.

Also - thank you for the tip on the debugging web ui. Now I can build out the rest of a kodi interface for basic setup of tvh!

#8 - 2017-05-13 17:48 - Jaroslav Kysela

[edit4ever !](#) : Don't use http requests for the kodi. All /api http requests are available through the htsp protocol / session, too. Just use "args" (map) and "path" (string) for the "api" method. The result is same htmsg structure as json result for http calls.